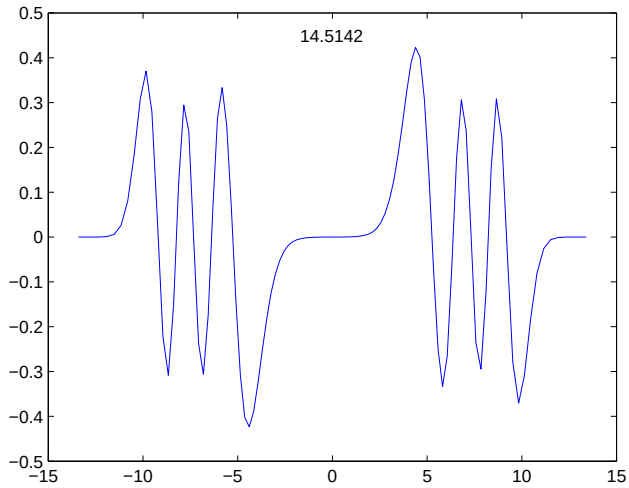
HermiteEig Tutorial

Damian Trif
dtrif@math.ubbcluj.ro

Department of Applied Mathematics
Babes-Bolyai University of Cluj-Napoca
ROMANIA



2006

1 The problem

The package `HermiteEig` approximates a finite number of eigenvalues and eigenfunctions of time-independent one-dimensional **Schrödinger** problem

$$-\frac{d^2u}{dx^2} + V(x)u = Eu, \quad x \in (-\infty, \infty), u \in L^2(-\infty, \infty)$$

Introducing the linear transformation $\xi = \alpha x$ for $\alpha > 0$ the eigenvalue problem becomes

$$-\frac{d^2\Psi}{d\xi^2} + \alpha^{-2}V(\alpha^{-1}\xi)\Psi = \mathcal{E}(\alpha)\Psi, \quad \xi \in (-\infty, \infty), \ \Psi \in L^2(-\infty, \infty) \tag{1}$$

where the eigenvalues $\mathcal{E}(\alpha)$ depending on the parameter α are connected with the original eigenvalues E by $E = \alpha^2\mathcal{E}(\alpha)$.

By setting

$$\Psi(\xi) = e^{-\xi^2/2}y(\xi),$$

(1) becomes

$$-y'' + 2\xi y' + \mathcal{V}(\xi)y = \mathcal{E}(\alpha)y$$

where

$$\mathcal{V}(\xi) = \alpha^{-2}V(\alpha^{-1}\xi) - \xi^2 + 1$$

is the modified potential.

We search an approximation of the new unknown function y of the form

$$y_N(\xi) = \sum_{n=0}^{N-1} c_n\psi_n(\xi)$$

where

$$\psi_n(\xi) = \frac{1}{\sqrt{2^n n!\sqrt{\pi}}}H_n(\xi), \quad n = 0, 1, 2, ...$$

are the normalized Hermite polynomials.

For the practical implementation, we define the vectors c and t by

$$c^T = (c_0, c_1, c_2, ...)$$
$$t^T = (\psi_0, \psi_1, \psi_2, ...)$$

so that $y(\xi) = c^T t(\xi) = t^T(\xi)c$.

Let us find now a matrix X for which

$$\xi \cdot y(\xi) = \xi \cdot c^T t(\xi) = (Xc)^T t(\xi),$$

i.e. the coefficients of $\xi \cdot y((\xi)$ are Xc. By using the property of Hermite polynomials

$$H_{n+1}(\xi) = 2\xi H_n(\xi) - 2nH_{n-1}(\xi)$$

we have for the functions ψ_n the three-term recursion

$$\xi\psi_n(\xi) = \sqrt{\frac{n}{2}}\psi_{n-1}(\xi) + \sqrt{\frac{n+1}{2}}\psi_{n+1}(\xi)$$

where $\psi_{-1}(\xi) = 0$ and $\psi_0(\xi) = \pi^{-1/4}$. Consequently, the non-zero elements of X are

$$X_{i,i+1} = X_{i+1,i} = \sqrt{\frac{i}{2}}, \quad i = 1, 2, ...$$

Of course, in practical calculation we must truncate this infinite matrix to have a finite order N., so that we obtain only an approximation of the exact result. The corresponding Matlab command of the package is

```
X=mult(N);
```

Then, in general,

$$\xi^m y(\xi) = (X^m c)^T t(\xi)$$

and

$$f(\xi)y(\xi) = (f(X)c)^T t(\xi)$$

for analytical functions f, i.e.

$$f(\xi) = \sum_{k=0}^{\infty} f_k \frac{\xi^k}{k!}$$

Moreover,

$$\frac{y(\xi)}{\xi^m} = \left(X^{-m}c\right)^T t(\xi)$$

if the l.h.s. has no singularity at the origin.

Similarly, let us find the differentiation matrix D giving

$$\frac{dy}{d\xi} = (Dc)^T t(\xi).$$

The derivatives of H_n satisfy $H_n' = 2nH_{n-1}$ so that $\psi_n' = \sqrt{2n}\psi_{n-1}$ and the non-zero elements of D are

$$D_{i,i+1} = \sqrt{2i}, \; i = 1, 2, ...$$

We truncate this matrix to the order N but this time, if y is described by N coefficients c_n, the above differentiation is exact. More, we have

$$\frac{d^2y}{d\xi 2} = (D^2c)^T t(\xi).$$

The Matlab command of the package to calculate the matrix D of order N is

```
D=deriv(N);
```

Applying these formulae to equation (1), we get

$$\left[-D^2 + 2XD - X^2 + \mathcal{V}(X) + I_N\right] c = \mathcal{E}c$$

If we define the matrix

$$A = -D^2 + 2XD - X^2 + \mathcal{V}(X) + I_N$$

the vector c satisfies the eigenvalues/eigenvectors problem

$$Ac = \mathcal{E}c \tag{2}$$

2 MATLAB codes

1) The package `HermiteEig` contains the files `pd.m`, `x2t.m`,`mult.m`, `deriv.m` and `sph.m`

a) The function `pd.m` (physical domain),

```
[x,w]=pd(n);
```

calculates the nodes and weights for the GaussHermite quadrature, i.e.

$$I \equiv \int_{-\infty}^{\infty} e^{-x^2} f(x)dx \approx \sum_{k=1}^{n} f(x_k)w_k = \texttt{w*f}$$

b) The function `x2t.m`

```
t=x2t(n,x)
```

calculates the values of the normalized Hermite polynomials ψ_k, $k = 0, 1, ..., n$ at the points x.

Consequently, if

$$f(x) = c_0\psi_0(x) + c_1\psi_1(x) + ... + c_n\psi_n(x)$$

and $c^T = (c_0, ..., c_n)$, we have $f(x) = c^T * t = t^T * c$, for $x \in (-\infty, \infty)$.

The code `x2t.m` could be used to transform the spectral representation c of the function f into the physical representation $f(x)$ of values at the grid x,

$$f(x) = t' * c$$

or to transform the physical representation $f(x)$ into the spectral representation c

$$c = \left(t^{-1}\right)' * f(x)$$

for moderate values of n. Of course, the direct formula

$$c_n = \int_{-\infty}^{\infty} e^{-x^2} f(x)\psi_n(x)dx$$

together with `pd.m` could be used, i.e.

$$c = [t * diag(f)] * w^T$$

For non-analytical functions f, the formula $f(\xi)y(\xi) = (f(X)c)^T t\,(\xi)$ doesn't work. In his case, we can use as transformation matrix

$$F = [t * diag(w)] * \left[diag\,(f(\xi)) * t^T\right]$$

instead of $f(X)$.

c) The functions `X=mult(n)` and `D=deriv(n)` calculate the matrices X respectively D.

d) The main function `[L,psi,x,w,t]=sph(fun,n,a)` calculates the eigenvalues L, the eigenfunctions psi, the nodes x, the weights w and the matrix t. The input parameters are the potential $V(x)$ in the function fun, the dimension of the discretized problem n and the scaling factor a. Of course, only the first eigenvalues are highly accurate while the last ones are spurious.

The subfolder `examples` contains some test problems from the literature, many of them from MATSLISE [1] and SLEDGE [5].

3 Examples

3.1 Morse problem

Let us consider the potential [2],

$$V(x) = (e^{-\gamma x} - 1)^2, \quad \gamma = 0.005\pi$$

with $N = 100$ and the optimization parameter $\alpha = 0.1$. Many of he calculated eigenvalues (file `morse.m`, command `morse`) coincide with the exact eigenvalues

$$E_n = \left(n + \frac{1}{2}\right)\gamma\left[2 - \left(n + \frac{1}{2}\right)\gamma\right], \quad n = 0, 1, ..., \left[\frac{1}{\gamma} - \frac{1}{2}\right]$$

see the figure 1.

Other example is the problem [1]

$$V(x) = 9e^{-2x} - 18e^{-x}$$

The numerical computation (`morse1`) gives, for $N = 100$ and $a = 1$

```
L(1:4)
ans =
 -6.25001304171899
 -2.24999748945656
 -0.24999051632800
 0.09169739225792
```

This problem has a continuous spectrum in $(0, \infty)$, with only 3 negative eigenvalues: -6.25, -2.25 and -0.25 so that the above result is $1.e-5$ accurate. However, the truncation of the domain to a bounded interval is more efficient.

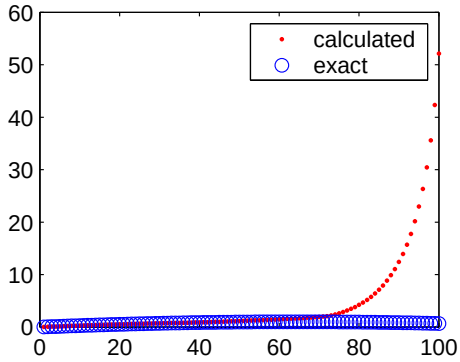


Figure 1:

3.2 Symmetric double well potential

Let us consider the potential [2]

$$V(x) = \upsilon_4 \left(x^2 - \frac{1}{2}\upsilon_4^{-1} \right)^2$$

with $\upsilon_4 = 0.01$, $\alpha = 1.1$ and $N = 100$. The calculated eigenvalues (file `sdwp.m`, command `sdwp`) are

```
L(1:12)
ans =
  1.40404860529767
  1.40404860529776
  4.17019360599931
  4.17019360599934
  6.87008883371401
  6.87008883371409
   9.49857838718786
  9.49857838719109
 12.04930948633407
 12.04930948667311
 14.51420502298122
 14.51420504812104
```

in very good coincidence with those from [2]. See also figure 2 for a plot of the eigenfunction corresponding to $L(12)$ eigenvalue.

3.3 Polynomial potential

Let us consider the quartic potential (file `quartic.m`, command `quartic`)

$$V(x) = x^2 + \upsilon_4 x^4$$

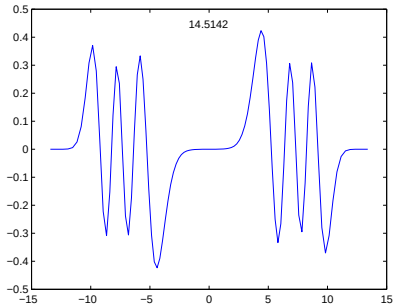


Figure 2:

for $N = 100$, $\alpha = 1$ and $\upsilon_4 = 0.01$. The first calculated eigenvalue (ground state energy) is

```
L(1)
ans =
1.00737367208140
```

with a difference of $2.e - 14$ to that calculated in [3].

Other potential of this type is the sextic potential (file `sextic`)

$$V(x) = x^2 + \upsilon_6 x^6$$

For $N = 100$, $\alpha = 1.8$ and $\upsilon_6 = 0.01$, the first calculated eigenvalue (ground state energy) is

```
L(1)
ans =
1.01674136375471
```

with the same difference $2.e - 14$ to that calculated in [3].

3.4 Gauss potential

Let us take now

$$V(x) = -e^{-\beta x^2}$$

for $\beta = 0.01$, $\alpha = 0.3$ and $N = 100$. The first six calculated eigenvalues (`gauss`) are

```
L(1:6)
ans =
-0.90376398798077
-0.71916893344509
```

```
-0.55080167079856
-0.39974972170785
-0.26746369362934
-0.15601325257593
```

with differences less than $1.e-14$ to that calculated in [3].

3.5 Asymmetric double well potential

Let us consider the potential [3]

$$V(x) = c_1 x^2 (x + c_2)(x - 1)$$

for $c_1 = 100$, $c_2 = 0.25$, $\alpha = 1.2$ and $N = 128$. The first four calculated eigenvalues (`adwp`) are

```
L(1:4)
ans =
-4.27734483985478
7.08051743865770
19.81776188766494
36.20933855713141
```

with a difference $1.e-6$ to that calculated in [3].

3.6 Anharmonic potential

Let us consider the problem [1]

$$V(x) = x^2 + \frac{\lambda x^2}{1 + g x^2} \ , \lambda = 0.1, \quad g = 0.1$$

The numerical computation (`anharmonic`) gives, for $N = 100$ and $a = 1$ the first eigenvalue `L(1)=1.04317371304445,` which coincides with the value found by *matslise*.

3.7 Bender-Orszag potential

Let us consider

$$V(x) = -\frac{m(m+1)}{\cosh^2(x)}$$

The known eigenvalues are $E_k = -(m-k)^2$, $0 \le k < m$. The calculated values for $m = 3$, $a = 1$ and $N = 100$ are

```
-8.99999999998992
-4.00000000003926
-0.99999999993041
```

in good agreement with the exact values.

3.8 Non-analytical potential

Let us consider [5]

$$V(x) = \begin{cases} 1 - e^{-x^2}, & x < 0 \\ x\log(x+1), & x > 0 \end{cases}$$

with the unique eigenvalue $\lambda_0 = 0.7125276014$. In this case, the transformation matrix is `V=t*diag(w)*(diag(fun(x/a))*t');`

The command `[L,psi,x,w,t]=tab(250,1.35);L(1:2)` gives the numerical eigenvalues

```
0.71252779084313
1.04274875016349
```

and the first of them approximates the true eigenvalue with an error of $2x10^{-7}$.

3.9 Evolution problems

In [4] an instability was observed numerically when a kind of Hermite spectral method was applied for linear diffusion equation in a unbounded domain. Precisely, let us consider the problem

$$\begin{aligned} u_t - u_{xx} &= f(x,t), \quad x \in \mathbb{R}, \mathrm{t} > 0 \\ u(x,0) &= u_0(x), \quad x \in \mathbb{R} \end{aligned}$$

where

$$f(x,t) = \left[5(1+x)\cos 5(x+t) + \left(25 + 0.5 - 0.25x^2\right)\sin 5(x+t)\right] e^{-\frac{x^2}{4}}$$

and

$$u_0(x) = \sin 5x \; e^{-\frac{x^2}{4}}$$

such that the exact solution of the problem be

$$U(x,t) = \sin 5(x+t) \; e^{-\frac{x^2}{4}}.$$

With the transformation

$$u(x,t) = y(x,t)e^{-\frac{x^2}{2}}$$

the above problem becomes

$$\begin{aligned} y_t &= y_{xx} - 2xy_x + (x^2-1)y + f(x,t)e^{\frac{x^2}{2}}, \quad x \in \mathbb{R}, \mathrm{t} > 0 \\ y(x,0) &= u_0(x)e^{\frac{x^2}{2}}, \quad x \in \mathbb{R} \end{aligned}$$

If we expand

$$y(x,t) = \sum_{k=0}^{N} c_k(t)\psi_k(x)$$

we obtain finally for the coefficients c the differential system

$$c' = \mathcal{D}c + F$$
$$c(0) = c^0$$

where F and c^0 are the coefficients of the expansions of $f(x,t)e^{\frac{x^2}{2}}$ and $u_0(x)e^{\frac{x^2}{2}}$ with respect to ψ and $\mathcal{D}$ is the matrix $D^2 - 2XD + X^2 - I_N$.

Now, by using the Crank-Nicolson method, we obtain for $c^j = c(t_j)$ the equations

$$\frac{c^{j+1} - c^j}{dt} = \mathcal{D}\frac{c^{j+1} + c^j}{2} + \frac{F^{j+1} + F^j}{2}$$

i.e

$$\left(I - dt\frac{\mathcal{D}}{2}\right) c^{j+1} = c^j + \frac{dt}{2}\left(\mathcal{D}c^j + F^{j+1} + F^j\right)$$

Using $N = 64$ and $dt = 0.01$ (`diffusion.m`) , we obtain after 100 time steps the coefficients C of the numerical solution at $t = 1$ and the corresponding values $\left(t^T C\right) e^{-x^2/2}$, with a maximal error 4×10^{-5} at the grid points x. Using $t = 0.001$ and 1000 time steps the error becomes 4×10^{-7}, without any numerical instability.

3.10 Fokker-Planck equation

We test now the package for the solutions of the Fokker-Planck equation [6],

$$\frac{\partial f(x,t)}{\partial t} = \frac{\partial^2 \left(B(x)f(x,t)\right)}{\partial x^2} + \frac{\partial \left(A(x)f(x,t)\right)}{\partial x}$$

via eigenfunction expansion. It is more convenient to work with a self-adjoint operator. Let us consider the equilibrium solution

$$f_0(x) = \frac{1}{B(x)} \exp\left(-\int^x \frac{A(y)}{B(y)} dy\right).$$

If $\Phi(x,t)$ is defined by $f(x,t) = f_0(x)\Phi(x,t)$, we obtain the equation for Φ

$$\begin{aligned}\frac{\partial \Phi(x,t)}{\partial t} &= B(x)\frac{\partial^2 \Phi(x,t)}{\partial x^2} - A(x)\frac{\partial \Phi(x,t)}{\partial x} \\ &= -L\Phi(x,t)\end{aligned}$$

The operator L satisfies the eigenvalue equation

$$L\Phi_n(x) = \epsilon_n \Phi_n(x)$$

and it is a self-adjoint operator on the space spanned by the eigenfunctions Φ_n, with the inner product with the weight function $f_0(x)$. The function $\Phi(x,t)$ can be expanded by the complete set of eigenfunctions according to

$$\Phi(x,t) = \sum_n b_n e^{-\epsilon_n t} \Phi_n(x)$$

where the expansion coefficients b_n are determined by the initial condition $f(x,0)$.

Computationally it is more convenient to solve the above eigenvalue problem using the Schrödinger equation

$$-\frac{d^2\Psi_n(z)}{dz^2} + V(z)\Psi_n(z) = \epsilon_n \Psi_n(z)$$

where

$$\Psi_n(z) = \left(f_0(x(z))\sqrt{B(x(z))}\right)^{1/2}\Phi_n(x(z))$$
$$z(x) = \int^x (B(y))^{-1/2}\, dy$$

The effective potential is given by

$$V(z) = \frac{1}{4}\left[W^2(z) - 2W_z'(z)\right]$$

and

$$W(z) = \frac{1}{\sqrt{B}}\left(A + \frac{B'}{2}\right),$$
$$B' = \frac{dB(x(z))}{dz}$$

Let us consider the case of a the bistable system

$$\frac{\partial f(x,t)}{\partial t} = \epsilon\frac{\partial^2 f(x,t)}{\partial x^2} - \frac{\partial\left(\left(\gamma x - gx^3\right) f(x,t)\right)}{\partial x}$$

with $\gamma = g = 1$. The corresponding effective potential is

$$V(z) = \frac{\left(z^3 - z\right)^2}{4\epsilon^2} - \frac{1}{2\epsilon}\left(3z^2 - 1\right)$$

For $\epsilon = 0.0125$, the package (`evol.m` with $N = 101$, $\epsilon = 0.0125$, the scaling factor $a = 7$, the time step $dt = 0.1$ and the initial distribution $f(x,0)$ – a Gauss function) gives the eigenvalues

```
L(1:10)*49*0.0125
ans =
 -0.00000000000000
  0.00000000090974
  0.95897283387818
  1.81602047864552
  1.82507262410861
  1.83463714345201
  2.56075256468075
  3.00731102631005
  3.16316264504337
  3.42456872119340
 ................
```

and the time evolution (see figure 3), in good concordance with those from [7].

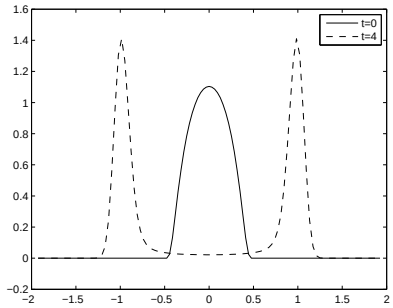


Figure 3: Fokker-Planck equation: time evolution

3.11 Complete list of examples

`adwp.m` asymmetric double well potential: $V(x) = c_1 x^2(x + c_2)(x - 1)$

`anharmonic.m` anharmonic potential $V(x) = x^2 + \frac{\lambda x^2}{1+gx^2}$

`bailey.m` Bailey potential, $V(x) = x^4 + 0.5x^3 - 7x^2$, weight $r(x) = 0.5$

`bender.m` Bender potential $V(x) = \frac{-42}{cosh^2(x)}$

`birkhoff.m` Birkhoff potential, $V(x) = x^4$

`bistable.m` bistable problem $u_t - \varepsilon u_{xx} + \left[(\gamma x - gx^3)u\right]_x = 0$, $u(x, 0) = u_0(x)$

`ch.m` hyperbolic cosine potential $V(x) = \cosh(x)$

`diffusion.m` diffusion problem $u_t - u_{xx} = f(x, t)$, $u(x, 0) = u_0(x)$

`fokker.m` Fokker-Planck operator $-\varepsilon u_{xx} + \left[(\gamma x - gx^3)u\right]_x = \lambda u$, $\varepsilon = 0.0125$, $\gamma = g = 1$

`gauss.m` Gauss potential: $V(x) = -e^{-\beta x^2}$

`harmonic.m` harmonic potential, $V(x) = x^2$

`marletta.m` Marletta potential, $V(x) = 8000t(t - 2)$ where $t = e^{-3x/2}$

`morse.m` Morse potential, $V(x) = \left(e^{-\gamma x} - 1\right)^2$

`morse1.m` Morse potential, $V(x) = 9e^{-2x} - 18e^{-x}$

`quartic.m` quartic potential: $V(x) = \nu x^4 + x^2$

`rat.m` rational potential, $V(x) = \frac{40x^2}{1+x^2}$

`razavy.m` Razavy potential $V(x) = \frac{m^2(cosh(4x)-1)}{8} - m(n+1)cosh(2x)$, $n = 2$, $m = 1, 10$

`sdwp.m` symmetric double well potential, $V(x) = \nu\left(x^2 - \frac{1}{2\nu}\right)^2$

`sextic.m` sextic potential: $V(x) = \nu x^6 + x^2$

`tab.m` $V(x) = 1 - e^{-x^2}$ for $x < 0$ and $V(x) = x\log(x + 1)$ for $x > 0$

`wolniewicz.m` Wolniewicz potential, $V(x) = 200(3x^4 - 6x^2 - 1)$

References

[1] V. Ledoux, MATSLISE package, `http://users.ugent.be/~vledoux/MATSLISE/`

[2] H. Taşeli, H. Alici, The scaled Hermite-Weber basis in the spectral and pseudospectral pictures, J. Math. Chemistry, 38,3, 367-378, 2005

[3] H. Taşeli, M. B. Erseçen, The scaled Hermite-Weber basis still highly competitive, J. Math. Chemistry, 34,3-4, 367-378, 177-187, 2003

[4] H. Ma, W. Sun, T. Tang, *Hermite Spectral Methods with a Time-Dependent Scaling for Parabolic Equations in Unbounded Domains*, SIAM J. Numer. Anal., 43,1,pp. 58-75, 2005

[5] S. Pruess, C.T. Fulton, Y. Xie, Performance of the Sturm-Liouville Software Package Sledge, T. R. MCS-91-19, 36 pp., September 1994

[6] G. W. Wei, *Discrete singular convolution for the solution of the Fokker-Planck equation*, J. Chem. Phys., 110, 18, pp. 8930-8942, 1999

[7] G. W. Wei, D. S. Zhang, D. J. Kouri, D. K. Hoffman, *Distributed approximating functional approach to the Fokker-Planck equation: Time propagation*, J. Chem. Phys., 107,8, pp 3239-3246, 1997